WFD Version 9 as on April 8, 2004

Command Summary

Function	R/W	Bits	Words	Description
Board Functions				
F0A8/F16A8	RW	16	1	Read/Write Board Control Register (BCR)
F0A9/F16A9	RW	16	1	Read/Write Xilinx Select Register (XSR)
F0A10/F16A10	RW	16	1	Read/Write JTAG Control Register (JCR)
F16A12	W	16	1	Write Data to memory
Channel Functions				
F0A0	R	24	32/2	Read Data from FIFO
F0A1	R	24	1536	Read Histograms
F0A2	R	24	1	Read Baseline
F0A3	R	24	16	Read Special Scalers
F1A0/F17A0	RW	16	512	Read/Write LUT
F1A1/F17A1	RW	24	1	Read/Wrire CSR
F1A2/F17A2	RW	16	1	Read/Write Trg
F1A3/F17A3	RW	16	1	Read/Write Win
F1A5/F17A5	RW	16	16	Read/Write Bunch Pattern
F1A6	R	16	1	Read Delimiter Counter
F1A7/F17A7	RW	16	1024	Read/Write Maximum/CFD Correction
F8A1				Copy Histograms to Memory
F9A0				Global reset
F9A1				Reset Scalers and Histograms
F9A2				Reset Address Pointers, Empty FIFOs
F9A3				Reset Revolution Counter
F9A6				Reset Delimiter Counter
Memory Controller Functions				
F0A0	R	16	49/4	Read Data from Memory
F0A1	RW	16	1	Read/Write Memory Controller CSR (MCSR)
F1A0/F17A0	RW	16	1	Read/Write Memory Pointer Lo16 (do this first)
F1A1/F17A1	RW	16	1	Read/Write Memory Pointer Hi16

BCR Summary (F0A8/F16A8, 16 bit)

Bit	R/W	Description
0	RW	Clock Select
1	R	0 – Reserved
2	RW	JTAG Select
3	W	Pulse 'PROG'
7:4	RW	BunchZ Delay
11:8	RW	Same Words to memory
14:12	R	0 – Reserved
15	R	'DONE' Status

CSR Summary (F1A1/F17A1, 24 bit)

Bit	R/W	Description
0:1	RW	Mode
3:2	RW	Integral Divider
4	RW	Enable Transfer to Memory
5	RW	Fine Histogram Mode
7:6	RW	Channel Number
8	RW	Enable 3-point Filter
9	RW	Enable Maximum Correction
10	RW	Enable 120-bunch Mode
11	RW	Enable 140 Mhz output (not used)
12	RW	Enable Rectangular Lookup
13	RW	Enable Internal Revolution Delimeter
14	R	Histogram to Memory Active
15	RW	Software Delimiter/Delimeter Pending
16	R	FIFO Overflow
17	R	Histogram Reset Active
18	R	Integral Overflow
19	R	Scaler/Hisogram Overflow
20	R	No Inhibit
23:21	R	0 – Reserved

Front Panel Summary

Name	Func.	Std.	Description
TTL A	Output	'LVTTL'	140 MHz monitor
TTL B	Output	'LVTTL'	Unused
TTL C	Output	'LVTTL'	Bunch, 15 CLK period, 1 CLK length, syncronized with Delayed Bunch Zero
TTL D	Output	'LVTTL'	Delayed Bunch Zero Monitor, 1800 CLK period, 1 CLK length
ACLK	Input	C-Decoupled	External Clock. $\pm$0.2 V is enough.
NIM B	Input	NIM	External Delimeter (GCC)
NIM C	Input	NIM	External Inhibit
NIM D	Input	NIM	External Bunch Zero
CH A	Input	Analog	Channel 0 analog input. $\approx$250 mV full scale.
CH B	Input	Analog	Channel 1 analog input
CH C	Input	Analog	Channel 2 analog input
CH D	Input	Analog	Channel 3 analog input
Offset	Jumper		Must be set to 'YES' position for negative signals
Offset	VarR		Baseline adjustment. Good number is 240 ampl. units for negative signals

Note: So called 'LVTTL' is 100 Ohm source terminated 3 V TTL output, making only 1 V on 50 Ohm load.

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td colspan="3">Board Functions A8-A9</td></tr>
<tr><td>Z,C</td><td>General Reset to the board and all channels</td><td></td></tr>
<tr><td>I</td><td>CAMAC Inhibit is ORed with the front panel Inhibit (Nim??) and is used for the blocking of data streams in all modes.
The resulting Inhibit is internally syncronized with BunchZ signal.</td><td></td></tr>
<tr><td>F0A8
F16A8
(16)</td><td>Read/Write Board Control Register (BCR):
<table>
<tr><th>Bit</th><th></th><th>Description</th></tr>
<tr><td>0</td><td>R/W</td><td>Board Clock Select (CAMAC interface is always clocked internally):
1 – External clocks from ACLK front panel input
0 – Internal 70 MHz clocks from crystal oscillator</td></tr>
<tr><td>1</td><td>R</td><td>0 – Reserved</td></tr>
<tr><td>2</td><td>R/W</td><td>Xilinx Chain JTAG select:
1 – JTAG controlled by CAMAC F0A10/F16A10
0 – JTAG repeated from/to back panel connector</td></tr>
<tr><td>3</td><td>W</td><td>1 – Pulse Xilinx 'PROG' – initiate Xilinx configuration from FLASH EEPROM</td></tr>
<tr><td>7:4</td><td>R/W</td><td>Delay of front panel BunchZ (Nim??) to channels in terms of 70 MHz clocks (=12 time units). 2 CLK are always added.</td></tr>
<tr><td>11:8</td><td>R/W</td><td>Number of the same words to be transferred to the Memory Controller with F16A12</td></tr>
<tr><td>14:12</td><td>R</td><td>0 – Reserved</td></tr>
<tr><td>15</td><td>R</td><td>Xilinx 'DONE' status: 1 – 'DONE'</td></tr>
<tr><td>23:16</td><td>R</td><td>0 – Not used</td></tr>
</table></td><td></td></tr>
<tr><td>F0A9
F16A9
(16)</td><td>Read/Write Xilinx Select Register (XSR).

On individual channel Write (F16–F23) or Control (F8–F15) functions all channels with corresponding XSR bit set will recieve the command and data.
On individual channel Read (F0–F7) functions data will be read from the channel with the lowest number for which the XSR bit is set.

<table>
<tr><th>Nw</th><th>23 16</th><th>15 5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>MemCntr</td><td>Ch3</td><td>Ch2</td><td>Ch1</td><td>Ch0</td></tr>
</table></td><td></td></tr>
<tr><td>F0A10
F16A10
(16)</td><td>Read/Write JTAG Control Register (JCR).
<table>
<tr><th>Nw</th><th>23 16</th><th>15 3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td></td><td>R</td><td>R</td><td>R</td><td>R/W</td><td>R/W</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>TDO</td><td>TDI</td><td>TMS</td></tr>
</table></td><td></td></tr>
<tr><td>F16A12
(16)</td><td>Write: transfer BCR[11:8]+1 words of data to memory. This is for test puposes only.</td><td></td></tr>
</table>

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td colspan="3">Individual Channel Functions A0-A7</td></tr>
<tr><td>F0A0
(24)</td><td>Read Data from data FIFO. Q=0 if no data availiable or if CSR4=1 (memory mode).

In 'WF' modes: 32 words per event:
<table>
<tr><th>Nw</th><th>23 16</th><th>15 8</th><th>7 0</th></tr>
<tr><td>0</td><td>Point2[7:0](B)</td><td>Point1[7:0](G)</td><td>Point0[7:0](R)</td></tr>
<tr><td>1</td><td>Point5[7:0](B)</td><td>Point4[7:0](G)</td><td>Point3[7:0](R)</td></tr>
<tr><td></td><td colspan="3">...</td></tr>
<tr><td>29</td><td>Point89[7:0](B)</td><td>Point88[7:0](G)</td><td>Point87[7:0](R)</td></tr>
<tr><td>30</td><td colspan="2">Revolution#[16:1]</td><td>R#[0] B#[6:0]</td></tr>
<tr><td>31</td><td>0</td><td colspan="2">Revolution#[32:17]</td></tr>
</table>
In 'AT' mode: 2 words per event:
<table>
<tr><th>Nw</th><th>23 16</th><th>15 8</th><th>7 0</th></tr>
<tr><td>0</td><td>Integral[7:0]</td><td>CFD Time[7:0]</td><td>Amplitude[7:0]</td></tr>
<tr><td>1</td><td>Rev#[8:1]</td><td>R#[0] B#[6:0]</td><td>Max Time[7:0]</td></tr>
</table>
In 'ALL' mode: 32 words per event; words 0–29 as in 'WF' modes, words 30–31 as in 'AT' mode.</td><td>CSR[1:0]=0,1;
CSR4=0

CSR[0:1]=2;
CSR4=0

CSR[0:1]=3;
CSR4=0</td></tr>
<tr><td>F0A1
(24)</td><td>Read Histograms: 1536 words. The format is:
<table>
<tr><th>Nw</th><th>Len</th><th>Description</th></tr>
<tr><td>0–119</td><td>120</td><td>Bunch# histogram.</td></tr>
<tr><td>120–127</td><td>8</td><td>Must read 0.</td></tr>
<tr><td>128–255</td><td>128</td><td>Amplitude histogram for unpolarized bunches. Bin width is 2 amplitude units, the histogram covers the whole range of amplitude values.</td></tr>
<tr><td>256–383</td><td>128</td><td>Same for positive polarization bunches</td></tr>
<tr><td>384–511</td><td>128</td><td>Same for negative polarization bunches</td></tr>
<tr><td>511–1535</td><td>1024</td><td>2-dim CFD Time vs Amplitude histogram. Dependent on CSR5 the histogram is:
<table>
<tr><th></th><th>Ch</th><th>BinW</th><th>Range</th></tr>
<tr><td colspan="4">CSR5=0 (Coarse)</td></tr>
<tr><td>Ampl</td><td>32</td><td>8</td><td>0:255, Full</td></tr>
<tr><td>Time</td><td>32</td><td>2</td><td>(WinB*2):(WinB*2+63),
≈1/3 of the whole range</td></tr>
<tr><td colspan="4">CSR5=1 (Fine)</td></tr>
<tr><td>Ampl</td><td>16</td><td>8</td><td>TrgH:(TrgH+127),
1/2 of the whole range</td></tr>
<tr><td>Time</td><td>64</td><td>1</td><td>(WinB*2):(WinB*2+63),
≈1/3 of the whole range</td></tr>
</table>
The readout is raw-wise, left to right, so that first come counts for sequential amplitude values for T=0, then for T=1 etc.</td></tr>
</table></td><td>CSR[1:0]=2
or
CSR[1:0]=3</td></tr>
</table>

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td>F0A2
(24)</td><td>Read current Baseline Average values. Mainly intended for test purposes. As the values are constantly changing and no latching is provided, there is no guarantee that the reading is always absolutely correct.
<table>
<tr><th>Nw</th><th>23 16</th><th>15 8</th><th>7 0</th></tr>
<tr><td>0</td><td>Mean2[7:0](B)</td><td>Mean1[7:0](G)</td><td>Mean0[7:0](R)</td></tr>
</table></td><td></td></tr>
<tr><td>F0A3
(24)</td><td>Read Special Scalers: 16 words in pairs (Lo,Hi) represent values of 8 32-bit scalers. Bits [19:16] of each word are the word number in the sequence. The current contents of all of the sclaers is latched on the reading of the Lo part of the first scaler, so that even while scaling is in progress, the values correspond to a single moment of time. Words 10–15 are not used, but must be read out to complete the sequence.
<table>
<tr><th>Nw</th><th>23 20</th><th>19 16</th><th>15 0</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>+Pol bunches through both cuts, [15:0]</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Same, [31:16]</td></tr>
<tr><td>2</td><td>0</td><td>2</td><td>−Pol bunches through both cuts, [15:0]</td></tr>
<tr><td>3</td><td>0</td><td>3</td><td>Same, [31:16]</td></tr>
<tr><td>4</td><td>0</td><td>4</td><td>0-Pol bunches through both cuts, [15:0]</td></tr>
<tr><td>5</td><td>0</td><td>5</td><td>Same, [31:16]</td></tr>
<tr><td>6</td><td>0</td><td>6</td><td>Filled bunches through AT cut, Int<LL, [15:0]</td></tr>
<tr><td>7</td><td>0</td><td>7</td><td>Same, [31:16]</td></tr>
<tr><td>8</td><td>0</td><td>8</td><td>Filled bunches through AT cut, Int>UL, [15:0]</td></tr>
<tr><td>9</td><td>0</td><td>9</td><td>Same, [31:16]</td></tr>
<tr><td>10</td><td>0</td><td>10</td><td>0</td></tr>
<tr><td>...</td><td>0</td><td>...</td><td>0</td></tr>
<tr><td>15</td><td>0</td><td>15</td><td>0</td></tr>
</table></td><td>CSR[1:0]=2
or
CSR[1:0]=3</td></tr>
<tr><td>F1A0
F17A0
(16)</td><td>Read/Write LookUp Tables (LUT), 512 words:
Words 0:255 represent limits for Time(Amplitude) lookup table, Lo byte – the lower limit for Time at the given Amplitude, Hi byte – the upper limit. Words 256:511 are the limits for the Integral(Amplitude) lookup table in the same manner.
<table>
<tr><th>Nw</th><th>23 16</th><th>15 8</th><th>7 0</th></tr>
<tr><td>0</td><td>0</td><td>T UL if A=0 [7:0]</td><td>T LL if A=0 [7:0]</td></tr>
<tr><td>1</td><td>0</td><td>T UL if A=1 [7:0]</td><td>T LL if A=1 [7:0]</td></tr>
<tr><td></td><td colspan="3">...</td></tr>
<tr><td>255</td><td>0</td><td>T UL if A=255 [7:0]</td><td>T LL if A=255 [7:0]</td></tr>
<tr><td>256</td><td>0</td><td>I UL if A=0 [7:0]</td><td>I LL if A=0 [7:0]</td></tr>
<tr><td></td><td colspan="3">...</td></tr>
<tr><td>511</td><td>0</td><td>I UL if A=255 [7:0]</td><td>I LL if A=255 [7:0]</td></tr>
</table>
Note: An event is considered to be good if the given parameter is strictly greater than AND strictly lower than the corresponding limit, so that values of Time and Integral 0 and 255 are always forbidden.</td><td></td></tr>
</table>

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td>F1A1
(24)
F17A1
(16)</td><td>Read/Write Control and Status Register (CSR):
<table>
<tr><td>0:1</td><td>R/W</td><td>Mode of operation:
0 – 'RAW' , signal waveform as is +B#/R#.
1 – 'SUBMEAN', waveform with baseline substraction and signal inversion +B#/R#.
2 – 'AT' , A/T/I/TMax/B#/R#, scalers active.
3 – 'ALL' , slow readout of waveform and calculated parameters, scalers active.</td></tr>
<tr><td>3:2</td><td>R/W</td><td>Integral divider: Calculated integral is divided by $2^{CSR[3:2]+2}$ in order to fit into a 8-bit value.</td></tr>
<tr><td>4</td><td>R/W</td><td>1 – Transfer data to memory as soon as it's ready.
0 – Direct FIFO readout</td></tr>
<tr><td>5</td><td>R/W</td><td>1 – Fine 2D histogram
0 – Coarse 2D histogram</td></tr>
<tr><td>7:6</td><td>R/W</td><td>0–3 – Channel Number. Should be programmed by the software to have the channel identification in the events stored to memory.</td></tr>
<tr><td>8</td><td>R/W</td><td>1 – Enable 3-point Filter</td></tr>
<tr><td>9</td><td>R/W</td><td>1 – Enable Maximum Correction</td></tr>
<tr><td>10</td><td>R/W</td><td>1 – 120 Bunch mode
0 – 60 Bunch mode</td></tr>
<tr><td>11</td><td>R/W</td><td>1 – Enable 140 MHz output (Not used any more)</td></tr>
<tr><td>12</td><td>R/W</td><td>1 – In 'AT' mode, use Rectangular Lookup instead of LUTs for the events going to FIFO or memory. No effect in other modes. No effect on scaler/histogram performance.</td></tr>
<tr><td>13</td><td>R/W</td><td>1 – Internal delimeter from revolution counter $/2^9$
0 – External delimeter from the front panel.</td></tr>
<tr><td>14</td><td>R</td><td>1 – Histogram transfer to memory active, should be 1 about 20 μs after F8A1 (never tested).</td></tr>
<tr><td>15</td><td>W

R</td><td>1 – Software delimeter (will immediately transfer the delimeter to memory if CSR4=1 and no INH).
1 – Delimiter pending.</td></tr>
<tr><td>16</td><td>R</td><td>1 – FIFO Overflow.</td></tr>
<tr><td>17</td><td>R</td><td>1 during Scaler Reset routine about 20 μs after F9A1.</td></tr>
<tr><td>18</td><td>R</td><td>1 – Integral Overflow: set if the integral $/2^{CSR[3:2]+2}$ exceeds 255.</td></tr>
<tr><td>19</td><td>R</td><td>1 – Scaler Overflow: set if one of the special scalers or histogram bins exceeds 2^{32} or 2^{24} correspondingly.</td></tr>
<tr><td>20</td><td>R</td><td>1 if event selection is not forbidden. Should reflect a NOR of crate INHIBIT and front panel INH.</td></tr>
<tr><td>23:21</td><td>R</td><td>– Reserved. Must read 0.</td></tr>
</table>
Note: Overflow bits (CSR16, CSR18, CSR19) are set once the error occurs and reset by writing new data to CSR.</td><td></td></tr>
</table>

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td>F1A2
F17A2
(16)</td><td>Read/Write Trigger and low limit Register (Trg):
Trg[7:0] – Low level trigger threshold in trems of RAW signal. In 'WF' modes the signal is put into the waveform FIFO and the baseline value is NOT calculated if the RAW signal has points lower than the TWR[7:0]. In other modes only used to block baseline calculations.
Trg[15:8] – Used as low limit in amplitude for:
2D histogram in Fine mode (CSR5=1)
Rectangular lookup window if enabled (CSR12=1)</td><td></td></tr>
<tr><td>F1A3
F17A3
(16)</td><td>Read/Write Window Register (Win):
Win[7:0] – Beginning of the sensitive window from BunchZ, in point frequency units, (1/6 of 70 MHz clock period = 2x time unit). Must never be set to 0. If even 0 is written, the value will be set to 1. Also 2x this value is used as low limit in time for:
2D histogram
Rectangular lookup window if enabled (CSR12=1)
Win[15:8] – End of sensitive window, same units. If set to $\geq$90, the window will remain open till the end of the bunch period.</td><td></td></tr>
<tr><td>F1A5
F17A5
(16)</td><td>Read/Write Bunch Pattern, 16 words. Two bits +Bn and –Bn are used to designate bunch polarity:
<table>
<tr><th>+Bn</th><th>–Bn</th><th>Bunch Signatire</th></tr>
<tr><td>0</td><td>0</td><td>Not Filled</td></tr>
<tr><td>1</td><td>0</td><td>Positive</td></tr>
<tr><td>0</td><td>1</td><td>Negative</td></tr>
<tr><td>1</td><td>1</td><td>Unpolarized</td></tr>
</table>
The format is:
<table>
<tr><th>Nw</th><th>23 16</th><th>15</th><th>14 . . . 9</th><th>8</th><th>7</th><th>6 . . . 1</th><th>0</th></tr>
<tr><td>0</td><td>0</td><td>–B7</td><td>. . .</td><td>–B0</td><td>+B7</td><td>. . .</td><td>+B0</td></tr>
<tr><td>1</td><td>0</td><td>–B15</td><td>. . .</td><td>–B8</td><td>+B15</td><td>. . .</td><td>+B8</td></tr>
<tr><td></td><td>0</td><td colspan="6">. . .</td></tr>
<tr><td>14</td><td>0</td><td>–B119</td><td>. . .</td><td>–B112</td><td>+B119</td><td>. . .</td><td>+B112</td></tr>
<tr><td>15</td><td>0</td><td colspan="6">X</td></tr>
</table></td><td></td></tr>
<tr><td>F1A6
(16)</td><td>Read current Delimeter Counter value.</td><td></td></tr>
<tr><td>F1A7
F17A7
(16)</td><td>Read/Write Maximum/CFD correction tables, 1024 words. Never has been seriously tested, but does no harm when tables are programmed with 0 and CSR9=1.
Each word of the Maximum correction table represents:
CORR[3:0] – addition MC to the defined maximum value if it is found at R point
CORR[7:4] – the same if found at G point
CORR[11:8] – the same if found at B point</td><td>CSR9=1</td></tr>
</table>

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td></td><td>10-bit address CADR[9:0] of the maximum correction data is formed of:
CADR[4:0] – signed difference DF (bits [6:2]) of the waveform values of the points, following and preceeding the point of maximum. If the difference doesn't fit in 5-bit, it's set to the highest possible positive or lowest possible negative 5-bit value
CADR[9:5] – most significant bits [7:3] DM of the defined maximum
The upper part CORR[15:12] of each word represents a CFD threshold correction. This number is subtracted from the corrected maximum value before dividing it by 4 to form the CFD threshold. Each of R, G and B points is compared to it's own corrected threshold. This correction should, of course, be proportional to the corrected maximum value CM[7:0] to make a real CFD.
So the format is:
<table>
<tr><th>Nw</th><th>15 12</th><th>11 8</th><th>7 4</th><th>3 0</th></tr>
<tr><td>0</td><td>CFD R CM=0</td><td>MC B DM=0 DF=0</td><td>MC G DM=0 DF=0</td><td>MC R DM=0 DF=0</td></tr>
<tr><td>1</td><td>CFD G CM=0</td><td>MC B DM=0 DF=1</td><td>MC G DM=0 DF=1</td><td>MC R DM=0 DF=1</td></tr>
<tr><td>2</td><td>CFD B CM=0</td><td>MC B DM=0 DF=2</td><td>MC G DM=0 DF=2</td><td>MC R DM=0 DF=2</td></tr>
<tr><td>3</td><td>X</td><td>MC B DM=0 DF=3</td><td>MC G DM=0 DF=3</td><td>MC R DM=0 DF=3</td></tr>
<tr><td>4</td><td>CFD R CM=1</td><td>MC B DM=0 DF=4</td><td>MC G DM=0 DF=4</td><td>MC R DM=0 DF=4</td></tr>
<tr><td>5</td><td>CFD G CM=1</td><td>MC B DM=0 DF=5</td><td>MC G DM=0 DF=5</td><td>MC R DM=0 DF=5</td></tr>
<tr><td>6</td><td>CFD B CM=1</td><td>MC B DM=0 DF=6</td><td>MC G DM=0 DF=6</td><td>MC R DM=0 DF=6</td></tr>
<tr><td>7</td><td>X</td><td>MC B DM=0 DF=7</td><td>MC G DM=0 DF=7</td><td>MC R DM=0 DF=7</td></tr>
<tr><td>...</td><td colspan="4">...</td></tr>
<tr><td>32</td><td>CFD R CM=8</td><td>MC B DM=1 DF=0</td><td>MC G DM=1 DF=0</td><td>MC R DM=1 DF=0</td></tr>
<tr><td>33</td><td>CFD G CM=8</td><td>MC B DM=1 DF=1</td><td>MC G DM=1 DF=1</td><td>MC R DM=1 DF=1</td></tr>
<tr><td>34</td><td>CFD B CM=8</td><td>MC B DM=1 DF=2</td><td>MC G DM=1 DF=2</td><td>MC R DM=1 DF=2</td></tr>
<tr><td>35</td><td>X</td><td>MC B DM=1 DF=3</td><td>MC G DM=1 DF=3</td><td>MC R DM=1 DF=3</td></tr>
<tr><td>...</td><td colspan="4">...</td></tr>
<tr><td>1020</td><td>CFD R CM=255</td><td>MC B DM=31 DF=28</td><td>MC G DM=31 DF=28</td><td>MC R DM=31 DF=28</td></tr>
<tr><td>1021</td><td>CFD G CM=255</td><td>MC B DM=31 DF=29</td><td>MC G DM=31 DF=29</td><td>MC R DM=31 DF=29</td></tr>
<tr><td>1022</td><td>CFD B CM=255</td><td>MC B DM=31 DF=30</td><td>MC G DM=31 DF=30</td><td>MC R DM=31 DF=30</td></tr>
<tr><td>1023</td><td>X</td><td>MC B DM=31 DF=31</td><td>MC G DM=31 DF=31</td><td>MC R DM=31 DF=31</td></tr>
</table></td><td></td></tr>
<tr><td>F8A1</td><td>Copy Histograms to memory. CSR14 reflects the status of the operation. Never tested.</td><td>CSR4=1</td></tr>
<tr><td>F9A0</td><td>General Reset to the channel: resets all registers, address pointers and register based counters to their initial values, which is usually 0. Does not affect the contents of the histograms.</td><td></td></tr>
<tr><td>F9A1</td><td>Reset Scalers: affects both histograms and special scalers. CSR17 reflects the status of the operation.</td><td></td></tr>
<tr><td>F9A2</td><td>Reset Address Pointers for all sequential operations such as: histogram and special scalers read, LUT read/write etc. Also resets both FIFOs to the empty state
In 'ALL' mode, this signal is generated internally after event readout, so that you can safely read out scalers and histos starting at address 0 after the event readout.</td><td></td></tr>
<tr><td>F9A3</td><td>Reset Revolution Counter. After this operation the counter will start counting revolutions only after the first allowing edge of INH combination.</td><td></td></tr>
<tr><td>F9A6</td><td>Reset Delimeter Counter. If the delimeter request is pending, it will also be reset.</td><td></td></tr>
</table>

Func.	Description	Condition
	Memory Controller Functions	
F0A0 F16A0 (16)	Read/Write data from/to memory, increment memory pointer. Q=1 always, so it's the responsibility of the software to keep track of the amount of data in the memory by reading out the pointer after the datastream is stopped. The event formats are:	
	In 'WF' modes: 49 words per event:	CSR[1:0]=0,1; CSR4=1; MCSR0=1

Nw	23 16	15 8	7 0
0	0	Channel CSR[15:0]	
1	0	Point1[7:0](G)	Point0[7:0](R)
2	0	Point3[7:0](R)	Point2[7:0](B)
3	0	Point5[7:0](B)	Point4[7:0](G)
	...		
45	0	Point89[7:0](B)	Point88[7:0](G)
46	0	Revolution#[8:1]	R#[0] B#[6:0]
47	0	Revolution#[24:9]	
48	0	0	Revolution#[32:25]

	In 'AT' mode: 4 words per event:	CSR[0:1]=2; CSR4=1; MCSR0=1

Nw	23 16	15 8	7 0
0	0	Channel CSR[15:0]	
1	0	CFD Time[7:0]	Amplitude[7:0]
2	0	Max Time[7:0]	Integral[7:0]
3	0	Rev#[8:1]	R#[0] B#[6:0]

	In 'ALL' mode: 49 words per event; words 0–45 as in 'WF' modes, words 46–48 as words 1–3 in 'AT' mode.	CSR[0:1]=3; CSR4=1; MCSR0=1
	In 'AT' and 'ALL' modes delimiters may appear in the datastream transferred to memory. The delimiter format is (2 words):	CSR[0:1]=2,3; CSR4=1; MCSR0=1

Nw	23 16	15 8	7 0
0	0	Channel CSR[15:0]	
1	0	Delimiter Counter[15:0]	

	If the software issues F8A1 command (never tested), the copy of the channel internal histograms will appear in the memory datastream. The format of the block is (3073 words):	CSR4=1; MCSR0=1

Nw	23 16	15 8	7 0
0	0	Channel CSR[15:0]	
1	0	Bin0 of Bunch# Histo [15:0]	
2	0	0	Bin0 of Bunch# Histo [23:16]
3–240	0	The rest of Bunch# Histo in such pairs	
241–256	0	0	
257–512	0	Amplitude Histo for unpolarized bunches	
513–768	0	Same for positive polarization bunches	
769–1024	0	Same for negative polarization bunches	
1025–3072	0	2D AT Histogram	

<table>
<tr><th>Func.</th><th>Description</th><th>Condition</th></tr>
<tr><td></td><td>Important Note: The type and size of each block of data is determined by the combination of bits in the first word of the block – CSR[15:0]:
<table>
<tr><th>CSR15</th><th>CSR14</th><th>CSR[1:0]</th><th>Len</th><th>Type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>49</td><td>Waveform in 'RAW' mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>49</td><td>Waveform in 'SUBMEAN' mode</td></tr>
<tr><td>0</td><td>0</td><td>2</td><td>4</td><td>Event parameters in 'AT' mode</td></tr>
<tr><td>0</td><td>0</td><td>3</td><td>49</td><td>Waveform and params in 'ALL' mode</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>2</td><td>Delimiter</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>3073</td><td>Histograms</td></tr>
</table></td><td></td></tr>
<tr><td>F0A1
F16A1
(16)</td><td>Read/Write Memory Controller CSR (MCSR):
<table>
<tr><td>0</td><td>R/W</td><td>1 – Enable Accepting Data.
0 – Data coming from channels is lost</td></tr>
<tr><td>1</td><td>R/W</td><td>1 – Enable readout caching. When this bit goes to 1 or a new memory pointer takes effect with F1A1, the corresponding SDRAM page is read to the readout cache. Allow $\approx 15\mu$s for this operation.
0 – Direct readout from SDRAM</td></tr>
<tr><td>2</td><td>R/W</td><td>1 – Enable FASTCAMAC L2 operation. Data on the CAMAC bus is changed so, that the controller latches it on both edges of CAMAC S1 strobe, except for the leading edge of the first S1. Only useful when MCSR1=1.
0 – Standard CAMAC operation</td></tr>
<tr><td>15:3</td><td>R/W</td><td>Reserved</td></tr>
</table></td><td></td></tr>
<tr><td>F1A0
F1A1
(16)</td><td>Read Lo16/Hi16 Memory Address Pointer. The current value of the pointer is latched on F1A0, so that the whole value always corresponds to a single moment of time, and the Lo16 part should be read first.</td><td></td></tr>
<tr><td>F17A0
F17A1
(16)</td><td>Write Lo16/Hi16 Memory Address Pointer. The value of the pointer takes effect on F17A1, so the Lo16 part should be written first with F17A0. The new SDRAM page is read to the readout cache on F17A1 if MCSR1=1. Allow $\approx 15\mu$s for this operation.</td><td></td></tr>
</table>

FIFOs

The depth of the event FIFO depends on the current mode:

Mode	Events	Words
'RAW'	16	512
'WF'	16	512
'AT'	256	512
'ALL'	1	32

In waveform modes FIFO is never overritten, events are not accepted if the FIFO is full. In this case next event gets into FIFO only after the earliest event was comletely read out.

In 'AT' mode the FIFO is always overriten even if it's FULL, so that if you are not in time with reading parameters, you will always accept the new ones. Yet precautions are taken that both parameter words read always belong to the single event.

In 'ALL' mode, events are not accepted during readout and several microseconds after it.

In all modes in case the FIFO is empty, Q-reply will be 0 on an attempt to read data. Naturally, address pointers are not incremented in this case.

If ever a FIFO overflow occurs, CSR16 is set and can only be cleared by General Reset or writing data to CSR.